VZDU # 36 MAY/JUNE 1992

DOWN UNDER CLUB

Editor
Harry Huggins
12 Thomas Str.
Mitcham. 3132
03-873-1408

Treasurer
Ron Allen
2 Orlando Str.
Hampton. 3188
03-598-4534

Here we are again, and it's nearly the middle of the year.

There will need to be a meeting in July, to judge the competition entries. I SUGGEST that be Sunday 5th July. It is open to all who intend to attend to phome me and we can make arrangements for a different date.

Once again I must "SCREAM" about the lack of contributations. This is a club, and I expect to hear from quite a few members in between N/Letters. It shouldn't be left to the few to keep it going. If you are competent with the VZ., write something for others to learn. If you are learning send us your problems. We can and will solve them for you, (I hope).

I have published articles on programming in FAST BASIC, SIMPLE BASIC, ASSEMBLER and Graphics. Surely you don't all understand it first-up. I don't.

There are those that always know any new aspect that is brought up. It is a pity they don't write an article for us, so we could share their knowledge. Look back over the past year. You will find there is only half-a-dozen names on the articles.

Even the High Scores have stopped coming in. They have got too high for starters. So I am reducing them to NIL again. I declare Peter Watson winner for this round. He will be barred from the games that he has set the high score. However he may still attempt to TOP the all time high scores that I have listed. When any of his present scores are equalled or surpassed, he is free to enter again. Unfortunately he has not a competitor, as McLeans have left the club.

You all saw the Scream Sheet in the last edition. All that went wrong was the Expansion Memory had packed up. And I thought that it was the flavour of the Chewing Gum was wrong. How wrong can you get?

AMAZING ADVENTURE
CONTINUED
BY PETER ROSS

```
3020 GOSUB6600:GOTO2900
3030 GOSUB10:PRINT"YOU GET IN THE CAR AND IT DRIVES";
3040 PRINT"OFF. THE DRIVER SAYS 'WHERE ARE"
3050 PRINT"YOU GOING.' WHAT DO YOU WANT TO"
3060 PRINT"DO? <SAY CASTLE> OR <SAY I DON'T";
3070 PRINT"KNOW>":INPUTA$
3090 IFA$="SAY I DON'T KNOW" THEN 3120
3100 IFA$="SAY CASTLE"THEN3380
3110 GOSUB6600:GOTO3030
3120 GOSUB10:PRINT"YOU SAY 'I DON'T KNOW?' BUT THE"
3130 PRINT"DRIVER REACHES INTO HIS POCKET "
3140 PRINT"AND PULLS OUT A GUN. THE DRIVER"
3150 PRINT"SAYS 'TELL ME THE TRUTH OR ELSE'";
3160 GOSUB6500
3170 PRINT"<SAY NO> OR <SAY YES>"
3180 INPUTA$
3200 IFA$="SAY NO"THEN3230
3210 IFA$="SAY YES"THEN3270:GOSUB6600:GOTO3120
3230 GOSUB10:PRINT"YOU SAY 'NO. NEVER.' THE DRIVER"
3240 PRINT"PULLS THE TRIGGER."
3250 PRINT"BANG..........................";
3260 GOTO 420
3270 GOSUB10:PRINT"YOU SAY 'YES. I AM GOING TO A"
3280 PRINT"CASTLE BEYOND A LARGE DOOR.' THE";
3290 PRINT"DRIVER SAYS 'THANK YOU' HE PULLS";
3300 PRINT"THE TRIGER."
3310 PRINT"BANG..........................";
3320 PRINT"PRESS RETURN TO CONTINUE";:INPUTA$
3330 GOSUB10:PRINT"THE JEWELS OF ORANUS WERE FOUND"
3340 PRINT"BY AN UNKNOWN PERSON. THOUGH "
3350 PRINT"ANOTHER ADVENTURER WAS SENT ON"
3360 PRINT"THIS MISSION BUT WAS NEVER EVER"
3370 PRINT"SEEN AGAIN.":GOTO420
3380 GOSUB10:PRINT"YOU SAY 'A LARGE CASTLE.' THE "
3390 PRINT"DRIVER SAYS 'I WILL BE DRIVING"
3400 PRINT"PAST A CASTLE BUT I DON'T KNOW"
3410 PRINT"IF ITS THE ONE YOUR LOOKING FOR"
3420 PRINT"BECAUSE THERE ARE THREE CASTLES"
3430 PRINT"IN THIS STATE.' THE DRIVER STOPS";
3440 PRINT"THE CAR AND SAYS 'IS THIS THE "
3450 PRINT"PLACE.':GOSUB6500
3460 PRINT"<SAY YES> OR <SAY NO>";
3470 INPUTA$
3490 IFA$="SAY NO"THEN3520
3500 IFA$="SAY YES"THEN3570:GOSUB6600:GOTO3380
3520 GOSUB10:PRINT"YOU SAY 'NO THANKS.' BUT A FEW"
3530 PRINT"SECONDS A TERRIBLE ACCIDENT
3540 PRINT"OCCOURS. A LARGE TRUCK SWERVES
3550 PRINT"IN FRONT OF YOU. SPLAT!.........";
3560 GOTO420
3570 GOSUB10:PRINT"YOU SAY 'YES' AND GET OUT OF THE";
3580 PRINT"CAR.THE CAR DRIVES OFF. YOU WALK";
3590 PRINT"OVER TO THE THE OLD CASTLE BUT"
3600 PRINT"IT HASN'T GOT A LARGE DOOR ALL"
3610 PRINT"IT HAS IS A SMALL WOOD DOOR WITH";
3620 PRINT"CHAINS AND LOCKS ALL OVER IT! "
3630 PRINT"YOU CAN SEE A BLUESTONE PATH TO"
3640 PRINT"YOUR LEFT AND THE DOOR TO YOUR"
3650 PRINT"RIGHT WHAT DO YOU WANT TO DO <GO";
3660 PRINT"LEFT> OR <GO RIGHT>";:INPUTA$
3680 IFA$="GO RIGHT"THEN3710
3690 IFA$="GO LEFT"THEN3800:GOSUB6600:GOTO3570
3710 GOSUB10:PRINT"GO WALK TO YOUR RIGHT AND TRY TO";
3720 PRINT"BREAK THE CHAINS. BUT JUST AS "
3730 PRINT"YOU BROKE ONE OF THE CHAINS A"
3740 PRINT"POLICE CAR DRIVES PAST. THEY SEE";
3750 PRINT"YOU BREAKING IN AND SENTENCE YOU";
3760 PRINT"TO DEATH!!!"
3770 SOUND1,9;1,9
3780 PRINT"YOU HAVE BEEN HANGED!!!"
3790 GOTO 420
3800 GOSUB10:PRINT"YOU DECIDE TO WALK TO YOUR LEFT.";
3810 PRINT"YOU ARE BY A VERY LARGE STONE "
3820 PRINT"WALL YOU CAN SEE SOME ROPE ON "
3830 PRINT"THE GROUND. WHAT DO YOU WANT TO"
3840 PRINT"DO? <EXAMINE ROPE> OR <GET ROPE>";
3850 INPUTA$
3870 IFA$="EXAMINE ROPE"THEN3900
3880 IFA$="GET ROPE"THEN3950:GOTO6600:GOSUB3800
3900 GOSUB10:PRINT"YOU EXAMINE THE ROPE. IT LOOKS"
3910 PRINT"LIKE IT IS VERY STRONG AND IT"
3920 PRINT"HASN'T HAD MUCH WEAR."
3930 PRINT"PRESS RETURN TO CONTINUE";:INPUTA$
3940 GOTO3800
3950 GOSUB10:PRINT"YOU PICK UP THE ROPE."
3960 GOSUB6500
3970 PRINT"<THROW ROPE> OR <CLIMB WALL>
3980 INPUTA$
4000 IFA$="CLIMB WALL"THEN4030
4010 IFA$="THROW ROPE"THEN4090:GOSUB6600:GOTO3950
4030 GOSUB10:PRINT"YOU CAN'T CLIMB THE WALL BECAUSE";
4040 PRINT"ITS TOO STEEP."
4050 PRINT
4060 PRINT"PRESS RETURN TO CONTINUE";
4070 INPUTA$
4080 GOTO 3960
4090 GOSUB10:PRINT"YOU THROW THE ROPE OVER THE WALL";
4100 GOSUB6500
4110 PRINT"<CLIMB ROPE> OR <SCALE ROPE>
4120 INPUTA$
4140 IFA$="CLIMB ROPE"THEN4170
4150 IFA$="SCALE ROPE"THEN4180
4160 GOSUB6600:GOTO4090
4170 GOSUB10:PRINT"YOU CLIMB THE ROPE AND JUMP OVER";:GOTO419
0
4180 GOSUB10:PRINT"YOU SCALE THE ROPE AND JUMP OVER";
4190 PRINT"THE WALL. YOU CAN SEE A SIGN."
4200 GOSUB6500
```

```
4210 PRINT"<READ SIGN> OR <KEEP WALKING>
4220 INPUTA$
4240 IFA$="READ SIGN"THEN4280
4250 IFA$="KEEP WALKING"THEN4350
4260 GOSUB6600
4270 GOSUB10:PRINT"YOU CAN SEE A SIGN.":GOTO4200
4280 GOSUB10:PRINT"THE SIGN SAYS:"
4290 PRINT"'KEEP WALKING FORWARD TILL YOU"
4300 PRINT"GET TO ANOTHER SIGN POST AND "
4310 PRINT"<CROSS> THE <BRIDGE>"
4320 PRINT
4330 PRINT"PRESS RETURN TO CONTINUE";:INPUTA$:
4340 GOSUB10:GOTO4200
4350 GOSUB10:PRINT"YOU KEEP WALKING FORWARD FOR A "
4360 PRINT"LONG TIME. YOU ARE NEAR A RUBISH";
4370 PRINT"BIN. YOU CAN SEE A PIECE OF OLD"
4380 PRINT"PAPER ON THE GROUND. WHAT DO YOU";
4390 PRINT"WANT TO DO? <READ PAPER> OR
4400 PRINT"<KEEP WALKING>";:INPUTA$
4420 IFA$="READ PAPER"THEN4450
4430 IFA$="KEEP WALKING"THEN4500:GOSUB6600:GOTO4350
4450 GOSUB10:PRINT"THE PIECE OF PAPER SAYS:"
4460 PRINT"BEWARE OF THE ELECTRONIC DOORS!"
4470 PRINT"AND THE EVIL ROBOTIC ARMS!"
4480 PRINT
4490 PRINT"PRESS RETURN TO CONTINUE";:INPUTA$:GOTO4350
4500 GOSUB10:PRINT"YOU KEEP WALKING FOR ABOUT THREE";
4510 PRINT"MINUTES. YOU HAVE NOW COME TO A"
4520 PRINT"ENORMOUS LAKE. WHAT DO YOU WANT"
4530 PRINT"TO DO? <SWIM LAKE> OR "
4540 PRINT"<WALK AROUND>
4550 INPUTA$
4560 Y
4570 IFA$="SWIM LAKE"THEN4600
4580 IFA$="WALK AROUND"THEN4700:GOSUB6600:GOTO4500
4600 GOSUB10:PRINT"YOU JUMP IN THE LAKE AND SWIM "
4610 PRINT"LIKE MAD BECAUSE THERE ARE MAN"
4620 PRINT"EATING PIRANAS AFTER YOU! BUT"
4630 PRINT"YOU ARE TO SLOW OF A SWIMER.":FORA=1TO5500:NEXT
4640 FORB=1TO8
4650 FORA=1TO2:POKE31063,60:X=USR(0):NEXT
4660 FORA=1TO7:NEXT
4670 FORA=1TO2:POKE31063,63:X=USR(0):NEXT
4680 FORA=1TO7:NEXT
4690 NEXTB:PRINT"CHOMP!!!":GOTO420
4700 GOSUB10:PRINT"YOU WALK AROUND THE LAKE. BUT "
4710 PRINT"YOU DON'T SEE ANYTHING."
4720 GOSUB6500
4730 PRINT"<KEEP WALKING> OR <EXAMINE LAKE>";
4740 INPUTA$
4760 IFA$="EXAMINE LAKE"THEN4800
4770 IFA$="KEEP WALKING"THEN4830:GOSUB6600:GOTO4700
4800 GOSUB10:PRINT"YOU EXAMINE THE LAKE BUT ALL YOU";
4810 PRINT"SEE IS MAN EATING PIRANAS!"
4820 GOTO4720
4830 GOSUB10:PRINT"YOU WALK PAST A TREE WHAT DO YOU";
4840 PRINT"WANT TO DO? <CLIMB TREE> OR
4850 PRINT"<EXAMINE TREE>";
4860 INPUTA$
4880 IFA$="CLIMB TREE"THEN4910
4890 IFA$="EXAMINE TREE"THEN5000:GOSUB6600:GOTO4830
4910 GOSUB10:PRINT"YOU CLIMB UP THE TREE. BUT LOTS"
4920 PRINT"OF BUGS AND INSECTS START "
4930 PRINT"CRAWLING AROUND YOU UNTILL YOUR"
4940 PRINT"WHOLE BODY IS COVERED IN THEM "
4950 PRINT"YOU CAN'T BREATH!"
4960 FORA=1TO10:POKE31063,50:X=USR(0):NEXT
4970 PRINT
4980 PRINT"PRESS RETURN TO CONTINUE";:INPUTA$
4990 GOTO430
5000 GOSUB10:PRINT"YOU EXAMINE THE TREE. YOU CAN "
5010 PRINT"SEE LOTS OF BUGS AND INSECTS IN "
5020 PRINT"THE TREE. YOU KEEP GOING FORWARD";
5030 PRINT"FOR ABOUT AN HOUR. YOU HAVE COME";
5040 PRINT"TO ANOTHER SIGN POST. BUT THE "
5050 PRINT"WRITING ON IT IS NOT VERY CLEAR";
5060 PRINT"YOU CAN SEE A BRIDGE TO YOUR "
5070 PRINT"RIGHT AND A SMALL DOOR TO YOUR"
5080 PRINT"LEFT.":GOSUB6500
5090 PRINT"<CROSS BRIDGE> OR <OPEN DOOR>
5100 INPUTA$:LOCK=RND(100)
5120 IFA$="OPEN DOOR"THEN5150
5130 IFA$="CROSS BRIDGE"THEN5210:GOSUB6600:GOTO5060
5150 GOSUB10:PRINT"YOU OPEN THE DOOR. BUT A MAD "
5160 PRINT"WOMAN COMES OUT WITH A CRICKET"
5170 PRINT"BAT! SHE HITS YOU ON THE HEAD!"
5180 FORB=1TO15:FORA=10TO15:POKE31063,A:X=USR(0):NEXTA:NEXTB
5190 FORB=1TO15:FORA=10TO15:POKE31063,A:X=USR(0):NEXTA:NEXTB
5200 GOTO420
5210 GOSUB10:PRINT"YOU CROSS THE BRIDGE"
5220 PRINT"YOU ARE NEAR A LARGE DOOR. YOU"
5230 PRINT"CAN SEE A WALLET ON THE GROUND."
5340 GOSUB6500:PRINT"<EXAMINE";
5350 PRINT"DOOR> OR <EXAMINE WALLET>
5360 INPUTA$
5370 IFA$="EXAMINE DOOR"THEN5410
5390 IFA$="EXAMINE WALLET"THEN5550:GOSUB6600:GOTO5220
5410 GOSUB10:PRINT"THERE IS A COMBINATION LOCK ON"
5420 PRINT"THE DOOR. WHAT DO YOU WANT TO DO";
5430 PRINT"<ENTER COMBINATION> OR <OPEN "
5440 PRINT"DOOR>";:INPUTA$
5460 IFA$="OPEN DOOR"THEN5490
5470 IFA$="ENTER COMBINATION"THEN5600
5480 GOSUB6600:GOTO5410
5490 GOSUB10:PRINT"YOU TRY TO OPEN THE DOOR BUT IT"
5500 PRINT"HAS AN........................";
5510 FORA=1TO100:B=RND(7):POKE31063,B:X=USR(0):NEXT
5520 PRINT"ELECTRIC SHOCK!"
5530 PRINT
5540 GOTO 420
5550 GOSUB10:PRINT"YOU EXAMINE THE WALLET. IT HAS"
5560 PRINT"A SMALL PIECE OF PAPER WHICH "
5570 PRINT"SAYS: ' COMBINATION "LOCK".'"
5580 PRINT
5590 PRINT"PRESS RETURN TO CONTINUE";:INPUTA$:GOSUB10:GOTO522
0
5600 GOSUB10:PRINT"ENTER COMBINATON";:INPUTA
5610 IFA=LOCK THENGOTO5650
```

```
5620 PRINT"THE WRONG COMBINATION. THE DOOR"
5630 PRINT"HAS A...........................";
5640 GOTO5510
5650 GOSUB10:PRINT"THE DOOR SWINGS OPEN.! YOU CAN"
5660 PRINT"NOW ENTER THE CASTLE. BUT THERE"
5670 PRINT"ARE TWO MORE DOORS TO OPEN!"
5680 GOSUB6500
5690 PRINT"<OPEN DOOR 1> OR <OPEN DOOR 2>
5700 INPUTA$
5720 IFA$="OPEN DOOR 1"THEN5750
5730 IFA$="OPEN DOOR 2"THEN5800
5740 GOSUB6600:GOTO5650
5750 GOSUB10:PRINT"YOU OPEN DOOR NUMBER 1. BUT YOU"
5760 PRINT"FALL INTO A DEEP CROCODILE PIT."
5770 FORA=1TO100:POKE31063,A:X=USR(0):NEXT
5780 PRINT"SPLAT...........................";
5790 GOTO420
5800 GOSUB10:PRINT"YOU OPEN DOOR NUMBER 2. WHICH IS";
5810 PRINT"THE RIGHT ONE. YOU CAN SEE A "
5820 PRINT"MACHINE GUN ON THE GROUND. YOU
5830 PRINT"CAN ALSO SEE A CASTLE WITH AN "
5840 PRINT"OPEN DOOR.WHAT DO YOU WANT TO DO";
5850 PRINT"<GET GUN> OR <ENTER CASTLE>"
5860 INPUTA$
5880 IFA$="ENTER CASTLE"THEN5910
5890 IFA$="GET GUN"THEN5940
5900 GOSUB6600:GOTO5800
5910 GOSUB10:PRINT"YOU TRY TO ENTER THE CASTLE BUT"
5920 PRINT"THE DOOR HAS A..................";
5930 GOTO5510
5940 GOSUB10:PRINT"YOU PICK UP THE MACHINE GUN WHAT";
5950 PRINT"DO YOU WANT TO DO? <FIRE GUN> OR";
5960 PRINT"<ENTER CASTLE>";:
5970 INPUTA$
5990 IFA$="ENTER CASTLE"THEN5910
6000 IFA$="FIRE GUN"THEN6020
6010 GOSUB6600:GOTO5940
6020 GOSUB10:PRINT"YOU FIRE THE GUN AT THE CASTLE"
6030 PRINT"SHELLS FLYING EVERYWHERE. AFTER"
6040 PRINT"10000 ROUNDS OF AMMUNITION THE"
6050 PRINT"CASTLE FALLS DOWN IN A LARGE"
6060 PRINT"HEAP. YOU CAN SEE A LARGE BOX "
6070 GOSUB6500
6080 PRINT"<FIRE AT BOX> , <UNLOCK BOX> OR"
6090 PRINT"<OPEN BOX>";:INPUTA$
6110 IFA$="OPEN BOX"THEN6150
6120 IFA$="FIRE AT BOX"THEN6200
6130 IFA$="UNLOCK BOX"THEN6250
6140 GOSUB6600:GOTO6020
6150 GOSUB10:PRINT"YOU TRY TO OPEN THE BOX BUT LOTS";
6160 PRINT"OF EVIL ROBOTIC ARMS COME OUT"
6170 PRINT"FROM IT AND TEAR YOU TO THREADS!";
6180 PRINT
6190 GOTO420
6200 GOSUB10:PRINT"YOU FIRE AT THE BOX BUT LOTS OF"
6210 PRINT"EVIL ROBOTIC ARMS COME OUT FROM"
6220 PRINT"IT AND TEAR YOU TO THREADS!"
6230 PRINT
6240 GOTO420
6250 GOSUB10:PRINT"YOU UNLOCK THE BOX WITH THE KEY"
6260 PRINT"YOU FOUND IN THE PIZZA SHOP. YOU";
6270 PRINT"OPEN THE LID AND FIND THE JEWELS";
6280 PRINT"OF ORANUS INSIDE. THE MAGICAL "
6290 PRINT"POWERS OF THE JEWELS TRANSPORT"
6300 PRINT"YOU BACK TO EARTH AND ITSELF"
6310 PRINT"BACK TO THE NATIONAL MUSEUM OF"
6320 PRINT"ORANUS."
6330 FORA=150TO1STEP-1:POKE31063,A:X=USR(0):NEXT
6340 GOSUB10:PRINT"WELL DONE YOU FOUND THE JEWELS"
6350 PRINT"OF ORANUS AND GOT TRANSPORTED "
6360 PRINT"BACK TO EARTH"
6370 PRINT:PRINT:PRINT
6390 PRINT"DO YOU WANT ANOTHER GO? <YES/NO>";
6400 INPUTA$
6410 IFA$="YES"THENRUN
6420 IFA$="NO"THENCLS:END
6500 PRINT"WHAT DO YOU WANT TO DO?":RETURN
6600 PRINT"SORRY. I DON'T UNDERSTAND":SOUND0,9
6610 INPUT"PRESS <<C>> TO CONTINUE.<<Q>> TO QUIT";B$
6620 IFLEFT$(B$,1)="Q"THEN6630ELSERETURN
6630 INPUT"ARE YOU SURE";C$:IFC$="Y"THEN150ELSERETURN
```

ASSEMBLER and the V.Z.ROM

by R.B.KITCH. Dec. 1991

A number of users have requested information on writing BASIC commands in Assembler. Routines already exist in the V.Z. ROM to carry out these functions. The entry points of the routines are difficult to discern, and interfacing Assembler programs is not well explained in V.Z. manuals and Z80 texts.

This article concentrates on interfacing the SOUND and COLOR commands, but the technique developed is widely applicable.

Assembler programmers should be impressed by the similarity between Assembler and Basic that exists when using the powerful routines that exist in ROM. Program development and debugging time is considerably reduced. Program size and execution time is also minimized.

Use of COLOR command from ASSEMBLER.

The question of setting foreground and background colors from Assembler raises a very interesting (I think!) discussion. It is one of the topics I have "earmarked" to write-up for V.Z. Users—but have not yet done so.

Discussion of BASIC ROM Routines.

The microsoft Level II ROM in the V.Z. contains a whole host of useful routines. Most are written to be interfaced into the BASIC Interpruter, but if called correctly they can be easily used by Assemblly language programmers. Some knowledge of the ROM and the interpreter is required to permit successful interfacing.

I will use the SOUND routine to illustrate my thinking, but will return to COLOR later.

BASIC action verbs, like SOUND and COLOR, are held as TOKENS in the BASIC program, followed by various parameters and punctuations expressed in ASCII (usually) and requiring a definite syntax for each command. The ROM contains the Basic Interpreter and Execution Driver at 1D1E h to 1D90 h. (disassemble this for your own interest.) The driver tests for the presence of a TOKEN, (>80 h), and then calculates an address in a jump table commencing at 1822 h to 1898 h.The interpreter then jumps off into the appropriate "action verb routines", that are located throughout the ROM. The action verb routine carry out two functions. The first, is syntax checking for the verb, and the second is carrying out the action. If the syntax is not correct, then control is passed to 1E4A h - the Functuin Code Error Handler.

An understanding of the above paragraph is vital to understanding the linking of BASIC verbs into Assembler programs.

From this discussion it would appear that there are at least THREE ways of linking Assembler with the ROM routines.

1. Create a "buffer" that contains a string of values resembling a Basic tokenised program, point the HL register at the start of the string, and then call the Interpreter at 1D1E h. This is simple to visualise, but error recovery can be difficult.

2. Create a "buffer" containing a string of parametere (and punctuation), point the HL register at the start, and call the action verb routine. This is the most straightforward and recommended approach to the particuler problem.(with COLOR)

3. Create an Assembler routine to achieve the desired result.The algorithm can be worked up from scratch or revised from the ROM's verb action table. This approach takes the most time,doesn't require ROM support, but can be very efficient, in terms of storage and speed. For example, to plot a line, but call the SET, RESET verb: as used by Basic, a suitable algorithm must be selected. Method 2. is the best one to follow, but as more facilities are required, then method 3. becomes useful. For example to clear the screen, (verb CLS) simply call D1C9 (the verb action routine). But suppose we want to clear to another color other than the default? Then a screen fill routine needs to be written from scratch. That will need a parameter indicating the desired screen color.

Essentially the Assembly Language programmer "fools" the ROM into thinking it is executing a BASIC program, when in fact it is being called from a machine language program, Herein lies the true power of the VZ's ROM routine. The ability of the Assembly language programmer relies upon his understanding of the V.Z. ROM to enable him to write very elegant programs.

Example using SOUND Verb Action Routine

The "BEEP" routine is located at 3450 h. and is very simple to call from Assembler as no papameters are required.(tone and duration are preset by "BEEP") The programmer may be required to save the working registers on the Z80 as they are all altered by "BEEP".

To make a beep in an Assembler program the following code is required:

```
PUSH AF              optional
PUSH BC              "     "
PUSH DE              "     "
PUSH HL              "     "
DI                   ; disable interrupt so that
                        duration is accurate
CALL 3450
EI                   (enable interrupt)
POP HL
POP DE
POP BC
POP AF
```

The disabling and later enabling of external interrupts is often omitted, but if accurate pitches (tones) and duration (m. sec) are required, then external time consuming events should be switched out.

A slightly more complex example is to call a "SOUND" having a certain pitch (HL reg,) and duration (BC reg.)

(I will omit the saving of registers in all examples from here on.)

```
LD BC, 75       ;pitch duration
LD HL, 259      ;duration
DI              ;switch out external events
CALL 345C       ;dedicate Z80 to looping through tone
                generating routine
EI              ;permit external events
```

Let us look at a more complex example of calling SOUND verb action routine at 2BF5 h. (haven't seen that address verb before for SOUND?). This time, the aim is to play a TUNE from assembler, i.e. a series of notes. The technique used is method 2 on the previous page of this article, i.e. fool the ROM into thinking it is executing some Basic program, held in a "BUFFER" area of RAM.

Firstly recall that the Sound command in BASIC specifies a pitch (0-31) and a duration (1-9) separated by a comma (,). There are actually 2 forms of syntax that will be accepted by the Interpreter. (although not necessarily described in the manual!).

FORM 1. SOUND #,#:SOUND #,# :SOUND #,# :

FORM 2. SOUND #,# ;#,# ;#,# ;#,#;.......:
where "#" is a 1-2 byte ASCII string of 30 h.-39 h.
(note; there are "parameters"—not like the "numbers" (0-255) previously put into BC and HL registers)

In the Basic program statement table (as seen by the interpretation and execution driver 1D1E h.) this would look like:

SOUND TOKEN,pitch ASCII number "," duration ASCII
number ";"........ ":"
9E h.,1-2 byte 30 h.-39 h.,2C h., 1 ASCII byte 30-39 h. 3B h.,3A h.

The execution driver detects the 9E h. token, and jumps off to the verb action routine at 2BF5 h. In this routine the syntax is checked, the ASCII range checked and converted to binary, and if O.K then the sound drivers at 3469 h. are called. If the syntax is incorrect then the error handling routines are entered at 1E4A h.

To achive all of this from Assembler and use the VZ's ROM routine involves the following piece of code.

TO PLAY A MELODY FROM ASSEMBLER

```
.
.
.
.
        LD HL,MELO          ;point to melody string
        CALL  2BF5          ;call sound verb action
                              routine
.                           ;continue assembler
                              program
```

```
        .
        .
        .
MELO    DEFM    "16,21;21,2;21,2;23,2;23,2;25,3;26,1;"
        DEFM    "28,2;26,2;25,2;23,2;23,2;21,4 "
        (Do you know the melody?)
```

The MELO label defines a data buffer section that is executed by the verb action routine as if it were part of the Basic program. On termination of the string, control is passed to the next Assembler instruction.

This technique is

very simple to program

very efficient in terms of RAM storage requirements

and quite fast in terms of execution speed.

All three criteria are highly desirable features for Assembler programs.

CALLING THE COLOR COMMAND FROM ASSEMBLER.

I trust that by now the method involved in connecting Assembler with Basic's ROM routine is becoming clear. Your mind should be running ahead at all of the possibilities that are available.

Let's first review the COLOR command in BASIC— like the SOUND command it has couple of parameters associated with it, and can be executed in a couple of forms.(syntax). The first parameter is the foreground color (1-8) and the second, is the background color (0-1). These colors vary depending upon the MODE (0\1, lo\hi, text\graphics) that the screen is set in. In mode(0)

The background color can be green(0) or orange (1). Eight foreground colors are available with the "block graphics" characters(80 h.-FF h.)—green, yellow, blue, red, buff, cyan, magenta, orange. In mode(1). The background colors are green (0) and buff (1). Four foreground colors are available depending upon the background color set. Color set (0) is green yellow blue and red. Color set (1) is buff cyan magenta and orange. A copy of the mode and background color selected is stored in the I/O latch located at 6800 h. and in a copy of the latch located at 783B h. in the communication area. Bit 3 is the display mode and Bit 4 is the background color in the latch. A copy of the foreground color currently set is located at 7846 h. also in the communication area. These are all areas of memory that control hardware - and also "remind" the Basic Interpretor of the currently set parameters.

After that long description of the COLOR command, you should have a fair idea of how to change the fore- and back-ground colors on the VZ display, from assembler. ie. write a routine from scratch.

I would recommend however using the verb action routine entered from 389D h. whenever the token 97 h. is interpreted by the execution driver. This ensures that a uniform approach to changing colors is taken. For example the display mode must be checked and the copy of the I/O latch at 783B h. must be updated whenever 6800 h. is changed. Assembly Language programmers sometimes "forget"to do all of the "housekeeping chores" required and unpredictable results follow.

The three forms of syntax associated with the color command are

FORM 1	COLOR F,B:	defines both d & b
FORM 2	COLOR , B:	defines b F is default at 7846 h.
FORM 3	COLOR F:	defines f. B IS default at 6300 h. and 783B h.

The interpreter sees this as

97 H.,ASCII 31-38 h.,2C h., ASCII 30-31 h.,3A h.

To reset foreground and background colors from Assembler the following code can be used.

```
        .
        .
        .
        .
        LD HL, COLS             ;point to color setting string
        CALL 389D h.            ;use color verb action routine
        .                       ;comtinue assembler program
        .
COLS    DEFM     "3,0"
```

FINALE.

This "brief" paper indicates how simple and concise it is to use the comprehensive range of routines present in the VZ Basic ROM. The approach suggested minimises the amount of debugging required, because the verb action routines have been verified by the Basic Interpreter. The Assembler interface proposed so closely follows Basic that debugging becomes trivial. "If it works in Basic it will work in Assembler". A considerable amount of mystery is removed from Assembler. Basic programers can easily extend their ideas into Assembler.

Supporting this paper are lists of Basic reserved words with corresponding tokens and verb action routine entry points that are VITAL to understanding my technique. Disassembled portions of the interpretation/execution driver and the COLOR verb action routine are provided for completeness.

I trust that a firm understanding of BASIC, Assembler and the ROM have resulted from working through this paper.

Color F,B VERB ACTION routine

(foreground, background)

```
389D 7E          LD    A,(HL)      ;GET NEXT CHARACTER
389E FE 2C       CP    ','         ;IS IT A COMMA?
38A0 28 20       JR    Z,38C2      ;YES FOREGROUND IS DEFAULT GOTO BACKGROUND
                                           DECODER

38A2 CD 1C 2B    CALL  2B1C        ;CONVERT ASCII TO HEX AND RETURN IN A REG.
                                           ERROR IF >255
38A5 B7          OR    A           ;SET ZERO FLAG IF 0—RANGE CHECK ON F
38A6 CA 4A 1E    JP    Z,1E4A      ;IF 0 THEN GOTO FUNCTION CODE ERROR ROUTINE
38A9 FE 09       CP    9           ;SET FLAGS FOR TEST AGAINST 9 —RANGE CHECK ON F
38AB D2 4A 1E    JP    NC,1E4A     ;IF >8 THEN GOTO FUNCTION CODE ROUTINE
38AE 3D          DEC   A           ;MAKE F IN RANGE OF 0-7 (RATHER THAN 1-9
38AF E6 07       AND   7           ;ZERO MSB OF F—TO ENSURE RANGE
38B1 CB 27       SLA   A           ;MULTIPLY F BY 16.-OR SHIFT LEAST SIG. NIBBLE
38B3 CB 27       SLA   A           ;TO MOST SIG. NIBBLE. ENSURE THAT BIT 7 IS 0
38B5 CB 27       SLA   A           ;OR THAT BITS 4,5,6,ARE CORRECTLY SET.(TO BE ANDed
38B7 CB 27       SLA   A           ;WITH BLOCK GRAPHICS TO PROVIDE COLOR)
38B9 32 46 78    LD    (7846),A    ;SAVE CORRECTED FOREGROUND COLOR IN COMM. AREA
38BC 7E          LD    A,(HL)      ;GET NEXT CHARACTER
38BD B7          OR    A           ;SET ZERO FLAG TO TEST FOR NULL.(END IF STATEMENT)
38BE C8          RET   Z           ;VERB ACTION FINISHED AS END OF BASIC STATEMENT
                                           REACHED
38BF FE 3A       CP    ':'         ;IS IT A COLON
38C1 C8          RET   Z           ;YES SO VERB ACTION FINISHED AS COMPOUNDED
                                           STATEMENT REACHED
38C2 CF          RST   8           ;TEXT NEXT.  BACKGROUND VALUE
38C3 2C          DEFB  ','         ;IS IT A COMMA? SYNTAX ERROR IF NOT
38C4 CD 1C 2B    CALL  2B1C        ;CONVERT ASCII TO HEX AND RETURN IN A REGISTER
                                           ERROR IF >255
38C7 B7          OR    A           ;SET ZERO FLAG TO TEST IF B=0 RANGE CHECK B
38C8 20 0C       JR    NZ,38D6     ;IF NOT ZERO GOTO TEST UPPER RANGE
38CA 3A 3B 78    LD    A,783B      ;COPY OF I/O LATCH IN COMMS.AREA PUT INTO A REG
38CD CB A7       RES   4,A         ;BIT 4 SET TO ZERO TO INDICATE BACKGROUND COLOR
38CF 32 3B 78    LD    (783B),A    ;UPDATE COPY OF I/O LATCH
38D2 32 00 68    LD    (6800),A    ; UPDATE I/O LAYCH
38D5 C9          RET               ;FINISHED



38D6 FE 01       CP    1           ;SET ZERO FLAG IF B=1
38D8 C2 4A 1E    JP    NZ,1E4A     ;B NOT EQUAL TO 1 SO GOTO FUNCTION CODE ERROR
38DB 3A 3B 78    LD    A,(783B)    ;COPY OF I/O LATCH IN COMMS. AREA PUT INTO A REG.
38DE CB E7       SET 4,A           ;BIT 4 SET TO 1 TO INDICATE BACKGROUND COLOR
38E0 32 3B 78    LD    (783B),A    ;UPDATE COPY OF I/O LATCH
38E3 32 00 68    LD    (6800),A    ;UPDATE I/O LATCH
38E6 C9          RET               ;FINISHED
```

VERB ACTION ROUTINES FOR TOKENS
80h – BBh :::: 128d – 187d

JUMP TABLE 1822h – 1899h
6178d – 6297d

80	END	1DAE	9E	SOUND	2BF5
81	FOR	1CA1	9F	RESUNE	1FAF
82	RESET	0138	A0	OUT	2AFB
83	SET	0135	A1	ON	1F6C
84	CLS	01C9	A2	OPEN	7979
85	CMD	7973	A3	FIELD	797C
86	RANDOM	01D3	A4	GET	797F
87	NEXT	22B6	A5	PUT	7982
88	DATA	1F05	A6	CLOSE	7985
89	INPUT	219A	A7	LOAD	7988
8A	DIM	2608	A8	MERGE	798B
8B	READ	21EF	A9	NAME	798E
8C	LET	1F21	AA	KILL	7991
8D	GOTO	1EC2	AB	LSET	7997
8E	RUN	1EA3	AC	RSET	799A
8F	IF	2039	AD	SAVE	79A0
90	RESTORE	1D91	AE	SYSTEM	0000
91	GOSUB	1EB1	AF	LPRINT	2067
92	RETURN	1EDE	B0	DEF	795B
93	REM	1F07	B1	POKE	2CB1
94	STOP	1DA9	B2	PRINT	2B6F
95	ELSE	1F07	B3	COMT	1DE4
96	COPY	3912	B4	LIST	2B2E
97	COLOR	3890	B5	LLIST	2B29
98	VERIFY	3738	B6	DELETE	2BC6
99	DEFINT	1E03	B7	AUTO	2008
9A	DEFSNG	1E06	B8	CLEAR	1E7A
9B	DEFDBL	1E09	B9	CLOAD	3656
9C	CRUN	372E	BA	CSAVE	34A9
9D	MODE	2E63	BB	NEW	1B49

DOS COMMAND ENTRY POINTS

VERSION 1.2A	VERSION 1.2B	
4A80	4A7C	REN COMMAND
4B08	4B04	INIT COMMAND
4D78	4D74	DRIVE COMMAND
4D92	4D8E	IN# COMMAND
4E64	4E5C	PR# COMMAND
4FFB	4FF3	DCOPY COMMAND
45DB	45DB	RUN COMMAND
48C4	48C4	BLOAD COMMAND
48EF	48EF	BRUN COMMAND

GAMES COLUMN

Welcome to the Games Column. This issue I shall talk about a few odds and ends, do a report on a printer game, and a review of a game by Scott La Brun.

***** STOP PRESS: Delete that about the review! A late visit to the Post Office revealed a letter from Peter Watson with reviews of his. So I'll keep mine for a later issue. *****

O.K. About three issues ago I reviewed the three games by Tom Thiel. Well I have been playing Road Warrior and I noticed that instead of the 'Radar' having two different coloured dots it had three. (White-You, Black-Bad guy, and BLUE-Fuel.), The problem was that my T.V went back to black and white.If you "eat" all the blue dots a message appears on the screen, "Fantastic! Get ready player one." But instead of getting a new level you get the same old one.

Now something for those of you with a printer --- CALENDAR!

I can not find the name of the programmer. The game is found on the second library tape (unless it has changed.). This programe "Will generate a calendar for any year in the range 1901-1999. All you have to do is specify the year." Before you press return to enter the date you must have the printer "on line" At the start of the printing you will get a "Snoopy" picture Under the picture is the year you want, in big figures, and then the months and days. This programe is useful if you decide to find the original day of your birthday or for just an ordinary date's calendar. Also there is a programme which can be used to double check "CALENDAR". This programe is also found on tape two of the library Games Tapes. It is called "BIRTHDAY" and will ask for your birthday and tell you which day you were born on.

Same advice for the more adventurous, if you would like to change the "Snoopy" picture, lines 820 to 1360 is the part to change to your own design. Also you can make a x-mas present with it. Give-away calendars are just not being given away anymore.

MAIL BAG ***** MAIL BAG ***** MAIL BAG *****

TWO REVIEWS BY PETER WATSON

CRASH-2 vs CRASH-4

In Crash-2 you have to run over the dots in the maze, at the same time avoiding the car that is coming the opposite way. This car can change into your lane, giving you only a fraction of a second to change lanes yourself. When you run over all the dots you start a new frame. I found this game very repetitive, and the ringing sound in the background gets annoying after a while.

Crash-4 is visually very similar, but there are a number of differences that make it quite different to play. Firstly, you only

have a limited fuel supply, so each frame doesn't last forever and a day. Also, in frame three (3) you get a second car trying to crash into you, and in frame five (5) you have three (3) cars all trying to stop you. To overcome this you have a speed button to get out of any tight spots. This makes Crash-4 a much varied game.

So, the final verdict is:

1st Crash-4 2nd Crash-2

CIRCUS VS BUST-OUT

In CIRCUS you have to move a jumping board to bounce the stick-figures up and burst the balloons. You also have to catch the figures as they fall or they will go splat on the ground. This game can be frustrating, as the figures do not always bounce in the direction that you want them to, and can fall in a place where it is impossible to catch them.

BUST-OUT is visually a much plainer game. The object, once again, is to burst balloons, but this time it is done with a bouncing ball. You must keep this ball from disappearing off the bottom of the screen by blocking its path with the bumper/platform. Although it is repetitive, it requires a certain amount of skill, whereas Circus relies more on luck (at least when I play.)

So I would rank them:

1st Bust Out 2nd Circus

Peter ends his letter:

"Oh, I also have a question. Have any members had the patience to see how long the VZ takes to make a move on Chess Level 6? I once tried it on level 5 with the expansion unit, and it took 50 minutes.

While I've got space here I'd like to say that I think Mitch's drawings in the past newssletters have been great, and I hope to see more in the future."

Mitch would like to say thanks for the compliment (thank you!) and I would like to say thanks for the reviews. (thanks they lighten my load.)

If you have some info you want printed, send to:

TIM PENDLEBURY
P.O.Box 917
Griffith, N.S.W. 2680.

Like the heading for the last issue this heading was done by my brother, Mitch, again using the "SKETCHES" joystick drawing program from the games library. I will use any headings sent to me.

And remember, all correspondence will be acknowledged in the games column, but if you want an earlier reply enclose a stamped addressed envelope with your letter and I will write to you.

THE HIGH SCORES

GAME	SCORE	LEVEL	HOLDER	ALL TIME HIGH	CHALLENGE	NAME
ASTEROIDS				110000		
BUST OUT				3940		
CIRCUS				3180		
CRASH-2	1969	2	PETER WATSON			
CRASH-4	881	4	PETER WATSON			
DAWN PATROL				99600	78100	PAUL FRANTZ
DEFENCE PENET	2086	J/S	TIM PENDLEBURY			
DEFENCE PENET	2643		TIM PENDLEBURY			
DIG OUT				83700		
GALAXON				339670	328460	PETER WATSON
GHOST BUSTER						
HAMBURGER SAM				99900		
HOPPY				25550		
KAMIKAZE	NEW A.T HIGH		************	109760	****174850	PETER WATSON
KILLER TOMATO	570	JS/KB ??	PETER WATSON	19900		
KNGS & DRAGONS		EASY		5300		
KNGS & DRAGONS		EXPERT		1200		
LADDER CHAL				29500	24380	PETER WATSON
LUNAR LANDER		TAPE 3		76600		
LUNAR LANDER		TAPE 5		8000		
MAZE OF ARGON				102684	78306	PETER WATSON
MISSILE ATTACK						
PANIK				16720	11540	PETER WATSON
PENGUIN				3610		
PHAROAH'S CURSE	204	1	TIM PENDLEBURY			
PHAROAH'S CURSE	297	2	TIM PENDLEBURY			
PHAROAH'S CURSE	321	3	TIM PENDLEBURY			
PHAROAH'S CURSE	255	4	TIM PENDLEBURY			
PHAROAH'S CURSE	171	5	TIM PENDLEBURY			
PLANET PATROL				1588		
SPACE RAM				1441		
STAR BLASTER	373	5	TIM PENDLEBURY			
STAR BLASTER	419	4	TIM PENDLEBURY			
STAR BLASTER	625	3	TIM PENDLEBURY			
STAR BLASTER	683	2	TIM PENDLEBURY			
STAR BLASTER	897	1	TIM PENDLEBURY			
SUPER SNAKE				1918		
TEN PIN BOWLS				255		
VZ INVADERS				30160		

PLEASE

When sending an article to Tim for inclusion in the NEXT newsletter,please remember that Tim has a deadline to meet with me. So please post early enough for Tim to get ir no later than the 10th of the month. Otherwise it may not be printed until the following month. (Ed.)

RESERVED WORD AND TOKENS LIST

1650h – 1821h IN ROM
5712d – 6177d

	DEC	HEX		DEC	HEX		DEC	HEX
END	128	80	(NAME)	169	A9	OR	211	D3
FOR	129	81	(KILL)	170	AA	>	212	D4
RESET	130	82	(LSET)	180	AB	=	213	D5
SET	131	83	(RSET)	181	AC	<	214	D6
CLS	132	84	(SAVE)	182	AD	SGN	215	D7
(CMD)	133	85	(SYSTEM)	183	AE	INT	216	D8
(RANDOM)	134	86	LPRINT	184	AF	ABS	217	D9
NEXT	135	87	(DEF)	185	B0	(FRE)	218	DA
DATA	136	88	POKE	186	B1	INP	219	DB
INPUT	137	89	PRINT	187	B2	(POS)	220	DC
DIM	138	8A	CONT	188	B3	SQR	221	DD
READ	139	8B	LIST	189	B4	RND	222	DE
LET	140	8C	LLIST	190	B5	LOG	223	DF
GOTO	141	8D	(DELETE)	191	B6	EXP	224	E0
RUN	142	8E	(AUTO)	192	B7	COS	225	E1
IF	143	8F	CLEAR	193	B8	SIN	226	E2
RESTORE	144	90	CLOAD	194	B9	TAN	227	E3
GOSUB	145	91	CSAVE	195	BA	ATN	228	E4
RETURN	146	92	NEW	196	BB	PEEK	229	E5
REM	147	93	TAB(	197	BC	(CVI)	230	E6
STOP	149	94	TO	198	BD	(CVS)	231	E7
ELSE	150	95	(FN)	199	BE	(CVD)	232	E8
COPY	151	96	USING	200	BF	(EOF)	233	E9
COLOR	152	97	(VARPTR)	201	C0	(LOC)	234	EA
VERIFY	153	98	USR	202	C1	(LOF)	235	EB
(DEFINT)	154	99	(ERL)	203	C2	(MKI$)	236	EC
(DEFSNG)	155	9A	(ERR)	204	C3	(MKS$)	237	ED
(DEFDBL)	156	9B	(STRING$)	205	C4	(MKD$)	238	EE
CRUN	157	9C	(INSTR)	206	C5	(CINT)	239	EF
MODE	158	9D	POINT	207	C6	(CSNG)	240	F0
SOUND	159	9E	(TIME$)	208	C7	(CDBL)	241	F1
(RESUME)	160	9F	(MEM)	209	C8	(FIX)	242	F2
OUT	161	A0	INKEY$	210	C9	LEN	243	F3
(ON)	162	A1	THEN	211	CA	STR$	244	F4
(OPEN)	163	A2	NOT	212	CB	VAL	245	F5
(FIELD)	164	A3	STEP	213	CC	ASC	246	F6
(GET)	165	A4	+	214	CD	CHR$	247	F7
(PUT)	166	A5	–	215	CE	LEFT$	248	F8
(CLOSE)	167	A6	*	216	CD	RIGHT$	249	F9
(LOAD	168	A7	/	217	CE	MID$	250	FA
			,	251	FB			

VERB ACTION ROUTINES FOR FUNCTION TOKENS
D7h - FAh :::: 215d - 250d

JUMP TABLE 1608h - 164Fh IN ROM
5640d - 5711d

D7	SGN	098A	E9	EOF	7961
D8	INT	0B37	EA	LOC	2964
D9	ABS	0977	EB	LOF	7967
DA	FRE	2704	EC	MKI$	7964
DB	INP	2AEF	ED	MKS4	796D
DC	POS	27F5	EE	MKD$	7970
DD	SQR	13E7	EF	CINT	0A7F
DE	RND	14C9	F0	CSNG	0AB1
DF	LOG	0809	F1	CDBL	0ADB
E0	EXP	1439	F2	FIX	0B26
E1	COS	1541	F3	LEN	2A03
E2	SIN	1547	F4	STR$	2836
E3	TAN	15A8	F5	VAL	2AC5
E4	ATN	15BD	F6	ASC	2A0F
E5	PEEK	2CAA	F7	CHR$	2A1F
E6	CV1	7952	F8	LEFT$	2A61
E7	CVS	7958	F9	RIGHT$	2A91
E8	CVD	795E	FA	MID$	2A9A

DOS ROUTINES -- ENTRY POINTS

	START ADDRESS		
JUMP TABLE	VERSION 1.2A	VERSION 1.2B	DESCRIPTION
4008	5F41	5F1B	DISK POWER ON
400B	5F52	5F2C	DISK POWER OFF
400E	4241	4241	ERROR HANDLING
4011	4717	4717	READ TRACK MAP
4014	4749	4749	CLEAR A SECTOR
4017	4754	4754	SAVE TRACK MAP
401A	4B08	4B04	INITIALISE DISK
401D	5367	535F	COMMAND INTERUPY
4020	53B9	53B1	ASCII TO HEX
4023	53EA	53E2	READ IDAM
4026	587B	5873	CREATE ENTRY
4029	58BF	58B7	FIND EMPTY SECTOR
402C	5913	590E	FIND ENTRY IN DIR
402F	5968	595C	FIND EMPTY SECT IN DIR
4032	59A1	5991	WRITE SECTOR
4035	5B27	5B17	READ SECTOR
4038	5EBE	5E98	DELAY M/SEC
403B	5ECE	5EA8	STEP IN
403E	5F01	5EDB	STEP OUT
4041	43B1	43B1	LOAD A FILE
4044	446E	446E	SAVE A FILE

PROGRAM INTERPRETER AND EXECUTIONER

EXECUTION DRIVER 1D5Ah. TO 1DE1h.

```
1D1E CD 58 03      CALL 0358         ;DOS EXIT & KEYBOARD AND SCANNER
1D21 B7            OR A              ;IF KEY WAS PRESSED ON SET FLAGS
1D22 C4 A0 1D      CALL  NZ,1DA0     ;SEE IF BREAK KEY WAS PRESSED
1D25 22 E6 78      LD    (78E6),HL   ;SAVE ADDRESS OF LAST BYTE EXECUTED IN CURRENT
                                          LINE
1D28 ED 73 EB 78   LD    (78EB),SP   ;SAVE CURRENT STACK POINTER
1D2C 7E            LD    A,(HL)      ;GET NEXT CHARACTER FROM INPUT STRING
1D2D FE 3A         CP    ':'         ;TEST FOR COMPOUND STATEMENT
1D2F 28 29         JR    Z,1D5A      ;JUMP IF COLON TO EXECUTION DRIVER
1D31 B7            OR    A           ;SET STATUS FLAGS
1D32 C2 97 19      JP    NZ,1997     ;NOT NUL BYTE.JUMP TO SYNTAX EXIT
1D35 23            INC   HL          ;POINT TO NEXT CHAR. IN STATEMENT (LSB)
1D36 7E            LD    A,(HL)      ;GET NEXT CHARACTER
1D37 23            INC HL            ;POINT TO MSB
1D38 B6            OR    (HL)        ;TEST FOR ZERO
1D39 CA 7E 19      JR    Z,197E      ;IMPLIES END OF STSTMENT. -EXIT
1D3C 23            INC HL
1D3D 5E            LD    E,(HL)      ;GET LINE NUMBER OF NEXT STATEMENT AND PUT
                                            IN DE
1D3E 23            INC HL
1D3F 56            LD    D,(HL)      ;DE = BINARY OF NEXT LINE NUMBER
1D40 EB            EX    DE,HL       ;HL = LINE NO. OF NEXT STATEMENT
1D41 22 A2 78      LD    (78A2),HL   ;UPDATE LAST EXECUTED LINE TO CURRENT LINE NO.
1D44 3A 1B 79      LD    A,(791B)    ;GET TRACE FLAG.  0=TROFF
1D47 B7            OR    A           ;SET STATUS FLAG
1D48 28 0F         JR    Z,1D59      ;JUMP IF TROFF. FALL THROUGH IF TRON
1D4A D5            PUSH  DE          ;SAVE DE SINCE DISPLAY USES IT
1D4B 3E 3C         LD    A,3C        ;ASCII ">"
1D4D CD 2A 03      CALL  032A        ;PRINT CHARACTER "A"
1D50 CD AF 0F      CALL  0FAF        ;CONVERT LINE NO. TO BINARY AND PRINT IT
1D53 3E 3E         LD    A,3E        ;ASCII "<"
1D55 CD 2A 03      CALL 032A       ;PRINT CHAR. IN A.  THIS GIVES LINE NO. IN TRON
1D58 D1            POP DE            ;RESTORE DE
1D59 EB            EX    DE,HL       ;HL = CODE STRING LINE NUMBER
1D5A D7            RST   10          ;GET NEXT TOKEN.
EXECUTION PHASE STARTS HERE
1D5B 11 1E 1D      LD    DE,1D1E     ;RETURN ADDRESS AFTER EXECUTING ONE VERB
1D5E D5            PUSH DE           ;SAVE RETURN TO STACK
1D5F C8            RET   Z           ;EXIT IF EOS (NULL) - GO BACK TO 1D1E
1D60 D6 80         SUB   80          ;COMPUTE RELATIVE TOKEN. (TOKEN RANGE 80-F6)
1D62 DA 21 1F      JR    C,1F21      ;NOT A TOKEN-MUST BE ASSIGNENT STATEMENT
1D65 FE 3C         CP    3C          ;TEST IF TOKEN BELOW TAB TOKEN
1D67 D2 E7 2A      JP    NC,2AE7     ;JUMP IF TOKEN =>BC (TAB-MID$)
1D6A 07            RLCA              ;TOKEN No. TIMES 2
1D6B 4F            LD    C,A         ;BC = ROUTINE OFFSET
1D6C 06 00         LD    B,0
1D6E EB            EX    DE,HL       ;SAVE HL (CURRENT LINE NUMBER)
1D6F 21 22 18      LD    HL,1022     ;POINT AT START VERB ACTION ROUTINES
1D72 09            ADD   HL,BC       ;HL=POINTER AT JUMP TABLE FOR VERB ACTION
                                                      ROUTINE
1D73 4E            LD    C,(HL)      ;PUT ADDRESS INTO BC
```

```
1D74 23         INC   HL          ;
1D75 46         LD    B,(HL)
1D76 C5         PUSH  BC          ;SAVE IT TO STACK
1D77 EB         EX    DE,HL       ;RESTORE CODE STRING ADDRESS
```

RESTART 10

```
Load and check next char. in string.   On entry HL points to start of string -1
 09, 0A(LF) and 20(SP) ignored         On exit A contains character
                                               HL points to character
                                               C=1 (if numeric)

1D78 23         INC   HL          ;BUMP THE NEXT CHARACTER
1D79 7E         LD    A,(HL)      ;GET NEXT CHARACTER
1D7A FE 3A      CP    ":"         ;COMPARE WITH COLON
1D7C D0         RET   NC          ;CHAR. > : SO FINISH. i.e it is a :,:,<,..A-Z
1D7D FE 20      CP    ' '         ;ELSE TEST FOR BLANK
1D7F CA 78 1D   JP    Z,1D78      ;GET NEXT CHAR. IF IT IS A BLANK
1D82 FE 0B      CP    0B          ;COMPARE WITH VERTICAL TAB
1D84 30 05      JR    NC,1D8B     ;JUMP IF A>=0B. NOT A CONTROL CODE
1D86 FE 09      CP    09          ;TEST FOR HORIZONTAL TAB
1D88 D2 78 1D   JP    NC,1D78     ;JUMP IF NOT A HORIZONTAL TAB (09) OR LF(0A)
1D8B FE 30      CP    '0'         ;COMPARE WITH ASCII ZERO
1D8D 3F         CCF               ;SET CARRY IF NUMERIC(>=30)
1D8E 3C         INC   A           ;CLEAR CARRY IF NOT NUMERIC (<30)
1D8F 3D         DEC   A           ;SET STATUS FLAGS (EXCEPT CARRY ) ACCORDING TO CHAR.
                                              JUST LOADED
1D90 C9         RET               ;RETURN TO CALLER
```

~~~~~~~~~~~ ATTENTION ~~~~~~~~~~~~

Peter Watson has asked that I publish his address, and to advise that he would like to correspond with other members. So don't be shy. He may let you into the secret of how he gets his HIGH SCORES. And you never know, between you perhaps I may get some good GEN. to help fill these pages.

Peter's address is:-

PETER WATSON, 3 GORDON STR., CULCAIRN. N.S.W. 2660

CORRECTION TO LAST ISSUE

When comparing IBM and VZ screens, I stated that VZ had 32 pixels to a screen. This was an error and should have been 128.

Thanks to Peter for drawing my attention to this. (Ed.)